



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Statistical abstraction for multi-scale spatio-temporal systems

Citation for published version:

Michaelides, M, Hillston, J & Sanguinetti, G 2017, Statistical abstraction for multi-scale spatio-temporal systems. in *International Conference on Quantitative Evaluation of Systems QEST 2017: Quantitative Evaluation of Systems*. Lecture Notes in Computer Science, vol. 10503, Springer, Cham, pp. 243-258, 14th International Conference on Quantitative Evaluation of Systems, Berlin, Germany, 5/09/17.
https://doi.org/10.1007/978-3-319-66335-7_15

Digital Object Identifier (DOI):

[10.1007/978-3-319-66335-7_15](https://doi.org/10.1007/978-3-319-66335-7_15)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

International Conference on Quantitative Evaluation of Systems QEST 2017

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Statistical abstraction for multi-scale spatio-temporal systems ^{*}

Michalis Michaelides¹,
Jane Hillston¹, and Guido Sanguinetti^{1,2}

¹ School of Informatics, University of Edinburgh

² SynthSys, Centre for Synthetic and Systems Biology, University of Edinburgh

Abstract. Spatio-temporal systems exhibiting multi-scale behaviour are common in applications ranging from cyber-physical systems to systems biology, yet they present formidable challenges for computational modelling and analysis. Here we consider a prototypic scenario where spatially distributed agents decide their movement based on external inputs and a fast-equilibrating internal computation. We propose a generally applicable strategy based on statistically abstracting the internal system using Gaussian Processes, a powerful class of non-parametric regression techniques from Bayesian Machine Learning. We show on a running example of bacterial chemotaxis that this approach leads to accurate and much faster simulations in a variety of scenarios.

1 Introduction

Modelling spatially extended dynamical systems is a task of central importance in science and engineering. Examples range from cyber-physical systems, to collective adaptive systems of human behaviour, to cellular systems. Despite their importance, computational modelling and analysis of such systems remains challenging due to a number of factors: the large number of degrees of freedom, the intrinsically hybrid nature of discrete systems existing in continuous space, and, frequently, the existence of multiple temporal scales in the system. As a result of these features, computational simulation of such systems is generally onerous, particularly in a stochastic setting [6,4].

In this paper, we consider the scenario where the system consists of multiple, spatially distributed, identical agents. The agents can sense an external, deterministic field and use this information to perform a stochastic, internal computation which determines the agent's subsequent move. The internal computation is often a system which will quickly reach a steady-state equilibrium when left unperturbed, e.g. a chemical reaction network. While this scenario is a special case as the agents do not interact with each other, it is sufficiently generic to cover many application scenarios, such as autonomous drones performing a task

^{*} M.M. and G.S. are supported by the European Research Council under grant MLCS 306999. J.H. is supported by the EU project, QUANTICOL 600708.

in space, or bacteria exploring a nutrient field. Such systems are cumbersome to handle computationally as the simulation of the internal computation needs to be repeated at every spatial step, so that simulating a single trajectory of the overall system may involve hundreds of simulations of the internal model.

Here we propose a novel approach to alleviate this computational burden based on emulating the statistics of the internal system. The central idea is to replace the expensive computation of the internal system with a lookup table which maps external stimulus to the output behaviour of the internal system. Crucially, we do not aim to model the detail of the internal state, but only an abstracted version capturing its qualitative behaviour (formalised as a logical property satisfied by the states). We achieve this by learning a parameters-to-behaviours regression map using Gaussian Processes (GPs), a powerful class of non-parametric Bayesian regression models. Our work is motivated by earlier work on using GPs to learn effective characterisations of system behaviour [2,1,11].

The rest of the paper is organised as follows: background on spatio-temporal systems and *E. coli* chemotaxis which serves as a running example (Section 2); the general framework for our statistical abstraction methodology, and its application to the chemotaxis system (Section 3); results assessing the quality and efficiency of the abstraction (Section 4); closing remarks about prospective expansion of the work (Section 5).

2 Background

2.1 Spatio-temporal agent models

We start by defining the class of spatio-temporal agent models we will consider in this paper. Let \mathcal{D} be a spatial domain (usually a compact subset of \mathbb{R}^n with $n = 2, 3$), and let $[0, T]$ be the temporal interval of interest. We define the *spatio-temporal field* $f: \mathcal{D} \times [0, T] \rightarrow \mathbb{R}$ to be a real-valued function defined on the spatial and temporal domains of interest. A spatio-temporal agent model is a triple $(\mathcal{D}, f, \mathcal{A})$ where \mathcal{A} is a collection of point *agents* whose location follows a stochastic process which depends on the spatio-temporal field. We note that this is not the most general case, as agents may be spatially extended, interact with each other or even influence the evolution of the spatio-temporal field. Nevertheless, such a level of abstraction is frequently adopted and justifiable in many practical applications.

Running example: chemotaxis in the Escherichia coli bacterium Foraging is a central problem for microbial populations. The bacterium *Escherichia coli* will normally perform a random walk within a spatial domain where nutrient concentration is constant (e.g. a Petri dish). When presented with a spatially varying nutrient field, a phenomenon known as *chemotaxis* arises. As the bacterium performs a random walk in the nutrient field encountering changing nutrient levels,

its sensory pathway effectively evaluates a temporal gradient of the nutrients (or ligands) it experiences; the walk is biased so that the bacterium experiences a positive temporal gradient more often than not [20,18]. Since the bacterium is moving in the field, the temporal gradient is implicitly translated into a spatial one, so the bacterium drifts toward advantageous concentrations. Implicitly translating a temporal gradient to a spatial one through motion is necessary for the bacterium cell, because its body size is too small to allow for effective calculation of the spatial gradient of a chemical field at its location. As a result, we can safely regard the bacteria as point-like agents.

2.2 Multi-scale models

In many practical situations, one is interested in modelling not only the movement of the agents, but also the mechanism through which sensing and decision making is carried out within each agent. This naturally leads to structured models with distinct layers of organisation, with behaviour in each layer informing the simulation that takes place at the layer above or below. We will assume that the internal workings of the agent are also stochastic, and we will model them as a *population Continuous Time Markov Chain* (pCTMC)³. Formally, a pCTMC is defined as follows.

Definition 1. *A population CTMC is a continuous-time Markov chain [12] with a discrete state-space \mathcal{X} , and an associated transition rate matrix Q . Each state in \mathcal{X} counts the number of entities of each type or “species” in a population, $\mathbf{X} \in \{\mathbb{N}^0\}^d$ for d species. Transitions in this space occur according to the rates given by Q .*

The transitions can be regarded as occurrences of chemical reactions, written as

$$\sum_{i=1}^d r_i X_i \xrightarrow{\tau(\mathbf{X})} \sum_{i=1}^d s_i X_i, \quad (1)$$

where for every species X_i , r_i particles of X_i are consumed and s_i particles are created. The transition rate $\tau(\mathbf{X})$ depends upon the current state of the system, and is the rate parameter of an exponential distribution governing the waiting times for this transition. The above transition rates of allowed reactions reconstruct the rate matrix Q .

Motor control in E. coli An *E. coli* cell achieves motility by operating *multiple* flagellum/motor pairs (F/M), which can either drive it straight (subject to small Brownian perturbation), or rotate it in place. Thus, the cell can either be ‘tumbling’ (re-orienting itself while stationary) or ‘running’ (propelling itself forward while maintaining direction) at any time (Figure 1: left, centre). The motility

³ The pCTMC is the internal model for a *single* agent here, not for multiple agents.

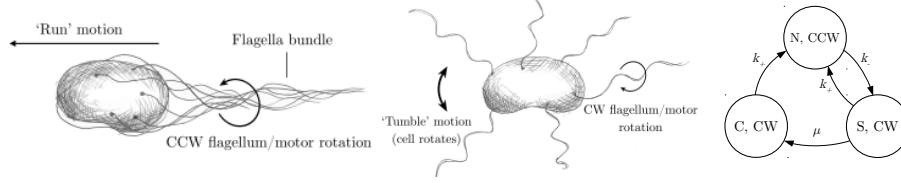


Fig. 1. The two motility modes of an *E. coli* cell. Left: the F/M are in CCW conformations, forming a helical bundle and propelling the cell. Centre: the F/M are in CW conformations, breaking the bundle apart and causing the cell to re-orient in place. Right: CTMC for a single F/M, with three conformation states and transition rates $k_{\pm}(m, L)$ and fixed $\mu = 5\text{s}^{-1}$.

state, RUN/TUMBLE, of the cell is determined by the number of flagella found in particular conformations. The model in [15] suggests three possible conformations for a flagellum: *curly* (*C*), *semicoiled* (*S*) and *normal* (*N*). The associated motor is modelled as a stochastic bistable system, which rotates either clockwise (CW) or counter-clockwise (CCW). Changes in motor rotation induce conformational changes on the associated flagellum. Transition rates between motor states are given by rate parameters k_+ and k_- for transitions $\text{CW} \rightarrow \text{CCW}$ and $\text{CCW} \rightarrow \text{CW}$, respectively. The possible transitions between flagellum/ motor states are summarised in the schematic diagram in Figure 1: right. *E. coli* normally has of the order of ten flagella and associated motors; the dynamics of the pair flagellum/ motor population therefore lends itself to be easily described as a pCTMC. The k_{\pm} transition rates depend on the temporal gradient evaluated by the chemotaxis pathway, and represent the functional interface of the bacterium with its external environment.

The classical mathematical model for the sensory response of the cell to external ligand concentration changes is provided by the Monod-Wyman-Changeux (MWC) model [17,15,9]. The model considers sensor clusters which signal information about ligand concentration changes to the motors, by triggering a biochemical response in the cell (phosphorylation of the CheY protein which binds to the motors) affecting the switching rates of rotation direction, k_{\pm} .

The full MWC model is still highly complex; in practice, we follow [15] and adopt a simplified model of sensory response to describe the dependency of motor rates k_{\pm} on ligand concentrations. This consists in abstracting the CheY signalling pathway in an effective variable m , which represents the methylation state of the ligand receptors and whose stochastic evolution is dependent on the ligand concentration L . Since m depends on past L concentrations the cell has been in, one may think of it as a *chemical memory* of sorts which encodes the value of L at previous times. The time comparison window is determined by how fast methylation happens — faster methylation leads to a shorter memory.

Sneddon et al [15] then resolve the entire dependency chain of the chemotaxis pathway to Equations 2 and 3. The motor switching rates $k_{\pm}(m, L)$ are given

by the deterministic equation

$$k_{\pm} = \omega \cdot \exp \left\{ \pm \left[\frac{g_0}{4} - \frac{g_1}{2} \left(\frac{Y_p(m, L)}{Y_p(m, L) + K_D} \right) \right] \right\}, \quad (2)$$

where

$$Y_p(m, L) = \alpha \cdot \left[1 + e^{\epsilon_0 + \epsilon_1 m} \cdot \left(\frac{1 + L/K_{\text{TAR}}^{\text{off}}}{1 + L/K_{\text{TAR}}^{\text{on}}} \right)^{n_{\text{TAR}}} \cdot \left(\frac{1 + L/K_{\text{TSR}}^{\text{off}}}{1 + L/K_{\text{TAR}}^{\text{on}}} \right)^{n_{\text{TSR}}} \right]^{-1}.$$

The methylation process can be naturally modelled as a birth / death process with rates depending on ligand concentration; again following [15] we take a fluid approximation of this, yielding the Ornstein-Uhlenbeck (OU) process:

$$\frac{dm}{dt} = -\frac{1}{\tau}(m - m_0(L)) + \eta_m(t). \quad (3)$$

In the above stochastic differential equation (SDE), $\eta_m = \sigma_m \sqrt{2/\tau} \Gamma(t)$, $\Gamma(t)$ is the normally distributed random process with 0 mean and unit variance, σ_m is the standard deviation of fluctuations in the methylation level, and $m_0(L)$ is an empirically derived function whose output is the methylation level required for full adaptation at the current external ligand concentration L . The adaptation rate τ , determines how fast methylation occurs and so, how long the ‘chemical memory’ of previous L values is in the system. The constants τ , along with $m b_0$ and α involved in the $m_0(L)$ function (see [15]), fully parametrise the methylation evolution. See [19] for reported values of constants used in Equation 2 and [5,15] for a detailed derivation of the results. Equations 2 and 3 couple the transition rates of the pCTMC in Figure 1: Right, with the external ligand concentrations, and therefore fully describe the internal model of the *E. coli* chemotactic response.

2.3 Simulating multi-scale systems

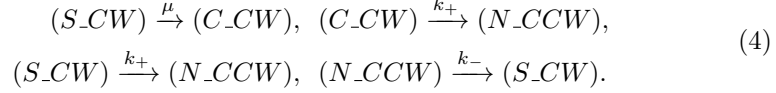
Multi-scale spatio-temporal systems are in general amenable to analytical techniques only in the simplest of cases. For the vast majority of real-world models, simulation-based analysis is the only option to gain behavioural insights.

Simulation of spatio-temporal systems typically employs nested algorithms: having chosen a time-discretisation for the spatial motion (which is assumed to have the slower time-scale), a spatial step is taken. Then, the value of the external field is updated, and the internal model is run for the duration of a given time-step with the new rates (corresponding to the updated value of the external field). A sample from the resulting state distribution then determines the velocity of the agent for the next time-step.

Clearly, this iterative procedure, while asymptotically exact (in the limit of small time discretisation), is computationally very demanding. This has motivated several lines of research in recent years [1,8,13,10].

Simulating chemotaxis in E. coli Simulations of the *E. coli* model outlined previously proceed along the general lines discussed above. Given a value of the ligand field and a characteristic time-step Δt , we draw samples of the SDE (3) using the Euler-Maruyama method, a standard method for simulating SDEs.

In the F/M pCTMC system in the reaction equation style, each species represents a different F/M conformation for a total of three species. The following transitions occur:



Note that in the above rate transitions there are dependencies on both external (L) and internal (m) states: $k_{\pm}(m, L)$, where L is an external input to the system (the external chemoattractant concentration at the time) and m is the current methylation level (sampled from the OU process in Equation 3 every Δt). Instead, the rate transition for $(S_CW) \rightarrow (C_CW)$ is fixed, $\mu = 5\text{s}^{-1}$.

Using the exact Gillespie algorithm [7], we then simulate the internal pCTMC for a length of time Δt to draw a sample configuration of the flagella/ motor system. Formally, trajectories of length Δt are checked against a property specifying the motility state for the cell (RUN/TUMBLE),

$$\phi_{\text{RUN}}(\mathbf{s}) = (N \geq 2) \wedge (S = 0), \quad (5)$$

where $\mathbf{s} = (S, C, N)$ is the last state of the flagella/ motor pairs in the CTMC trajectory.

The spatial location of the bacterium is then updated according to a simple rule: if the sampled internal state corresponds to RUN, the agent moves rectilinearly and updates its position $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{v} \cdot \Delta t$, where $v = 20\mu\text{m/s}$, the speed of the bacterium. Otherwise, if the internal state corresponds to TUMBLE, the agent remains still and its velocity is updated $\mathbf{v} \leftarrow R(\theta) \cdot \mathbf{v}$, where $R(\theta)$ is the standard 2D unitary rotation matrix through an angle θ , and θ is a tumbling angle sampled from a Gamma distribution as reported in Sneddon et al. [15].

The above simulation scheme, outlined in Algorithm 1, produces a chemotactic response to a ligand gradient. It takes $\sim 270\text{s}$ to simulate a single cell trajectory of $t_{\text{end}} = 500\text{s}$ with a time-step $\Delta t = 0.05$.

3 Methodology for statistical abstraction

In a multi-scale system, output from a set of processes in one layer in the system is passed as input to another layer; these processes are often computationally expensive. We present a methodology to abstract away such a set of processes and replace them with a more efficient stochastic map from the input to the output,

Algorithm 1 Simulation scheme for the *E. coli* model, based on full simulation of the pCTMC describing F/M conformation changes. Below, Δt is the fixed simulation time-step. Smaller functions called here (RUN, TUMBLE, OU-EULER-MARUYAMA) can be found in the full version of the algorithm, arXiv:1706.07005.

```

1: procedure SIMULATEFINECOLICELL( $t_{\text{end}}$ )
2:    $t \leftarrow 0$ 
3:   while  $t < t_{\text{end}}$  do
4:      $L \leftarrow L(\mathbf{r}, t)$  ▷ The ligand field  $L$  value, at the cell's
5:      $\mathbf{s} \leftarrow \text{PCTMC}(\mathbf{s}, m, L, \Delta t)$  location  $\mathbf{r}$ .
6:      $\psi \leftarrow \phi_{\text{RUN}}(\mathbf{s})$  ▷ Drawing F/M pCTMC trajectory of
7:     if  $\psi$  then length  $\Delta t$ , with parameters  $k_{\pm}(m, L)$ 
8:        $\mathbf{r} \leftarrow \text{RUN}(\mathbf{r}, \mathbf{v}, \Delta t)$  and initial state the last pCTMC
9:     else state of the cell.
10:       $\mathbf{v} \leftarrow \text{TUMBLE}(\mathbf{v}, \Delta t)$  ▷ Evaluating the  $\phi_{\text{RUN}}$  on (the final
11:    end if state of) the pCTMC trajectory.
12:     $m \leftarrow \text{OU-EULER-MARUYAMA}(m, L, \Delta t)$  ▷ Evolving methylation.
13:     $t \leftarrow t + \Delta t$ 
14:  end while
15: end procedure

```

governed by an underlying probability function. We approximate this probability function using Gaussian processes after observing many input-output pairs from the processes to be abstracted. The output consists of truth evaluations of properties expressed in logical formulae, which capture some behaviour of the system that is to be preserved by the abstraction.

3.1 Statistical abstraction framework

Consider a CTMC S , which given an initial state \mathbf{s}_0 , running time t , and input \mathbf{q} which completely determines transition rates, generates a trajectory $\mathbf{s}_{[0,t]}$. The trajectory is then checked for satisfaction of a property resulting in output $y = f(\mathbf{s}_{[0,t]}), y \in \{\top, \perp\}$. This layer of the multi-scale system can therefore be described as a set of operations:

$$S(\mathbf{s}_0, t, \mathbf{q}) = \mathbf{s}_{[0,t]}; \quad (6)$$

$$f(\mathbf{s}_{[0,t]}) = y. \quad (7)$$

Note that we consider a single property here for simplicity so a single binary value, but one could generalise to multiple properties, and hence, multi-valued output. This output then becomes input to a higher layer in the multi-scale system.

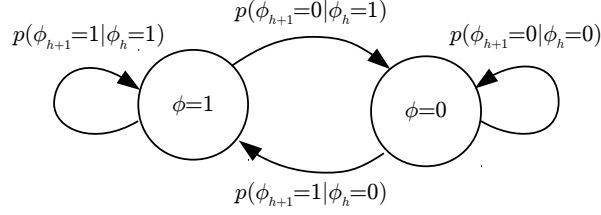


Fig. 2. DTMC with two states, $\phi_{\text{RUN}} \in \{\top, \perp\}$. The transition probabilities depend on internal methylation level m and external ligand concentration L .

Our goal is to construct a system \tilde{S} that is cheaper to simulate, whose output will be consistent with the original system S . Since the system is stochastic, in this context *consistent* refers to having the same probability distribution for the output random variable y . This abstracted system should generate output y' given the last output of the system y and the same input k as before:

$$\tilde{S}(y, \mathbf{q}) = y'. \quad (8)$$

Replacing the initial state \mathbf{s}_0 input with the previous output y allows us to substitute the whole layer of fine operations (6, 7) with the cheaper abstracted system \tilde{S} (8), in the multi-scale system. We regard this abstracted system to be a stochastic map from the internal state of the system and some given input to an output; we then use Gaussian processes to estimate the underlying probability function $\Psi(y, \mathbf{q})$ which governs the output of this stochastic map over the input domain.

Abstracting the E. coli chemotaxis pathway Returning to our model of the *E. coli* chemotaxis pathway, we associate the original system S with the pCTMC system of F/M conformations (Equations 4), along with the OU methylation process in Equation 3. The input starting state \mathbf{s}_0 is the last F/M state of the pCTMC, and the last methylation level m . The simulation time T is the variable Δt from Section 2.1, also used for the integration step-size of the OU in the Euler-Maruyama scheme. The transition rates k_{\pm} are calculated using the variables m and L , the last methylation level and external ligand concentration at the position of the cell, respectively. The output of this system, \mathbf{s}_t , is then a sampled pCTMC trajectory and new methylation level. Finally, the run property (5) is evaluated on (the last state of) the drawn pCTMC trajectory and the output determines whether the cell ‘runs’ or ‘tumbles’.

In observing the truth value of property ϕ_{RUN} for the state of the pCTMC at regular intervals of Δt , we cast the original pCTMC model (S) into a DTMC (Figure 2). This DTMC has only two states, $\phi_{\text{RUN}} \in \{\top, \perp\}$, and transition probabilities depending on the transition rates k_{\pm} , μ , of the original pCTMC.

Algorithm 2 Simulation scheme for the abstracted *E. coli* model, based on GP approximation for the RUN/TUMBLE probability. Steps 5, 6 here replace the expensive Steps 5, 6 in Algorithm 1.

```

1: procedure SIMULATEABSTRACTEDECOLICELL( $t_{\text{end}}$ )
2:    $t \leftarrow 0$ 
3:   while  $t < t_{\text{end}}$  do
4:      $L \leftarrow L(\mathbf{r}, t)$ 
5:      $p \leftarrow \text{GP}_{\psi}(m, L)$ 
6:      $\psi \sim \text{Bernoulli}(p)$ 
7:     if  $\psi$  then
8:        $\mathbf{r} \leftarrow \text{RUN}(\mathbf{r}, \mathbf{v}, \Delta t)$ 
9:     else
10:       $\mathbf{v} \leftarrow \text{TUMBLE}(\mathbf{v}, \Delta t)$ 
11:    end if
12:     $m \leftarrow \text{OU-EULER-MARUYAMA}(m, L, \Delta t)$ 
13:     $t \leftarrow t + \Delta t$ 
14:  end while
15: end procedure

```

Since this is only a two-state DTMC, the state at the next time-step conditioned on the current one can be modelled as a Bernoulli random variable:

$$\phi' \mid \phi \sim \text{Bernoulli}(p = p_{\phi'=1|\phi}(m, L)), \quad (9)$$

where ϕ, ϕ' are the ϕ_{RUN} DTMC states at time-steps $h, h + 1$ respectively. Also, the boolean $\{\perp, \top\}$ truth values of the properties have been mapped to the standard corresponding integers $\{0, 1\}$ for mathematical ease.

We recognise that a *single step* transition of this DTMC ($\phi' \mid \phi, m, L$) is the output $y' \mid y, \mathbf{q}$ produced by the abstracted layer $\tilde{S}(y, \mathbf{q})$. Identifying the corresponding probability function $p_{\phi'=1|\phi}(m, L)$ as the underlying governing function $\Psi(y, \mathbf{q})$ completes the setting of *E. coli* chemotaxis model abstraction to the methodology framework given above (Section 3.1). Note that the OU process for methylation is retained in the abstracted model as a parallel running process in the same layer of the multi-scale system. The OU process output m , together with the ligand concentration L (output of a different layer in the multi-scale system), constitute the input \mathbf{q} . The altered simulation scheme for this abstracted model is outlined in Algorithm 2. Notice how Steps 5, 6 there replace the more expensive Steps 5, 6 in Algorithm 1.

3.2 Approximating the underlying probability function Ψ

We use Gaussian process (GP) regression in order to infer the underlying probability function $\Psi(y, \mathbf{q})$ governing the stochastic \tilde{S} mapping from internal state y and input k to output y' in Section 3.1. A GP models a normally distributed

stochastic variable over a continuous domain. It can be thought of as a multivariate normal distribution over functions. This multivariate normal distribution can be conditioned on a finite number of (potentially noisy) observations of the function to be inferred, learning new mean and covariance parameters. These are computable at any point in the domain and correspond to the expected value of the function and associated variance at that point, respectively.

GPs are universal function approximators. The choice of covariance kernel determines the prior over the function and thus how many observations are required to get a good estimate of the underlying function. However, given enough observations, a GP with any valid kernel will approximate any smooth function arbitrarily well. We refer to [14] for a more comprehensive account of GPs.

Since training observations are binary samples of a Bernoulli distribution but GPs regress over a continuous unbounded variable, some adjustments must be made for correct evaluation of the underlying probability function Ψ . These are explained in the Gaussian process classification (GPC) method outlined in [14], and amount to identifying that class probability function with Ψ . We use Minka’s Expectation-Propagation (EP) technique to approximate the posterior because it is more accurate than the Laplace approximation. Further, we use fully independent training conditional (FITC) approximation [16] to allow a large number of observations to be considered for learning the underlying function, while maintaining a low cost of predicting at any point of the domain. Note that the Bernoulli distribution likelihood, used here for GPC, is a special case result because of both the binary $y = \phi$ output and the single observation of transitions at a particular (m, L) parametrisation.⁴ Lifting these restrictions would result in the more general multinomial distribution.

Constructing Ψ in E. coli chemotaxis As we mentioned, a single DTMC transition $(\phi' | \phi, m, L)$ corresponds to the output $y' | y, \mathbf{q}$ produced by the stochastic mapping $\tilde{S}(y, \mathbf{q})$. Therefore, $\tilde{S}(\phi, (m, L))$ consists of sampling from a Bernoulli distribution $\text{Bernoulli}(p = p_{\phi'=1|\phi}(m, L))$ where $p_{\phi'=1|\phi}(m, L)$ is the underlying probability function $\Psi(y = \phi, \mathbf{q} = (m, L))$ in the general formalism. We approximate $\Psi(y, \mathbf{q}) = p_{\phi'=1|\phi}(m, L)$, using GPs trained on observations from *micro-trajectories*, i.e. trajectories of the fine F/M pCTMC system which are then mapped onto the property space, $\phi \in \{0, 1\}$, to serve as training data.

Therefore, at a given (m, L) the pCTMC with transition rates $k_{\pm}(m, L)$ is at a state \mathbf{s}_0 which maps onto $\phi(\mathbf{s}_0)$. After a time Δt , the same CTMC is found at a state $\mathbf{s}_{\Delta t}$, which maps onto $\phi(\mathbf{s}_{\Delta t})$. An observation $\phi(\mathbf{s}_{\Delta t}) | \phi(\mathbf{s}_0), m, L$ is in this way recorded for every parametrisation (m, L) the bacterium has visited in the micro-trajectories.

Since the output of \tilde{S} is binary ($y = \phi \in \{0, 1\}$) we construct two probability functions $\Psi_{\phi}(m, L) = p_{\phi'=1|\phi}(m, L)$. Each is approximated with a separate GPC

⁴ It is highly unlikely to have more than a single transition since (m, L) are continuous values that constantly change for the bacterium.

function, where $\Psi_0(m, L)$ is trained on observations of transitions originating from the ‘TUMBLE’ state ($p_{\phi'=1|\phi=0}(m, L)$) and $\Psi_1(m, L)$ using transitions from the ‘RUN’ state ($p_{\phi'=1|\phi=1}(m, L)$). Notice that we need not estimate separate functions for $\phi' = \{0, 1\}$, since $p_{\phi'=1|\phi}(m, L) = 1 - p_{\phi'=0|\phi}(m, L)$. Having access to these underlying probability functions we are now able to sample the DTMC at any parametrisation (m, L) the bacterium finds itself in, by using the function estimate for $p_\phi(m, L)$ despite not having observations at that m, L .

The function $p_{\phi'=1|\phi}(m, L)$ is particularly challenging for GPs. This is due to a sharp boundary in the m, L domain, where there is a transition from $p_{\phi'=1|\phi}(m, L) \approx 0$ to $p_{\phi'=1|\phi}(m, L) \approx 1$. The bacterium has a steady state very close to this boundary, determined by the motor bias mb_0 , and that is where they are most often found. Therefore, accurate estimation of this boundary is crucial for this problem. Furthermore, the low probability of finding bacteria away from the boundary (in a relatively smooth ligand field) gives a very narrow window of where the function is observed. To get a better overall estimate, we sporadically perturb the position of bacteria in the micro-trajectory phase of collecting observations, such that the bacterium finds itself producing observations away from the boundary for a while, before the system returns close to steady state again. Despite these difficulties, we produce a good reconstruction of the underlying functions $p_{\phi'=1|\phi=0}$ and $p_{\phi'=1|\phi=1}$ over the m, L domain (see Figure 3).

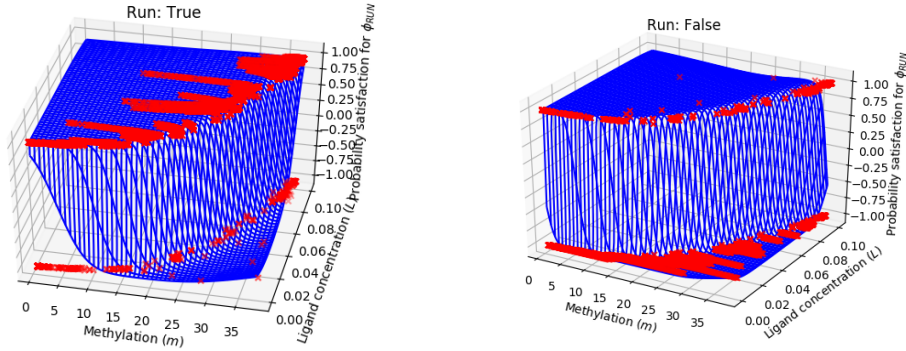


Fig. 3. The probability functions $p_{\phi'=1|\phi}(m, L)$ (left: $\phi = \top$, right: $\phi = \perp$) produced by the GP with hyperparameters $\ln(\ell) = (3.5, -2.5)$ and $\ln(\sigma) = 5$, 100 inducing points (FITC approximation), and 10000 observations (red crosses). The steep boundary is accurately captured producing a sharp switch-like transition from the run domain to the tumble domain.

4 Results

When assessing performance of our method for statistical abstraction, there are two things of interest: accuracy and computational savings. Accuracy refers to how similar behaviour of the abstracted system is to the behaviour of the original system. In our case of chemotaxis in *E. coli*, this is seen by comparing population distributions in a ligand field, resulting from simulations using the original fine system and the abstracted one. We also compare run and tumble duration distributions as another metric of how closely we approximate the output and behaviour of the original model.

Learning the transition probability functions for the dual-state DTMC enabled us to simulate bacteria using our abstracted model on a host of different ligand field profiles. Beyond comparing bacteria population distributions under the original Gaussian ligand field used for learning (see L_1 below), we did the same for a linear and dynamic field (L_2 , L_3 below), using the same learned functions $p_{\phi'=1|\phi}(m, L)$, $\phi \in \{0, 1\}$.

The ligand fields tested were:

$$L_1(\mathbf{r}) = 0.1 \cdot \exp \left[-0.5(\mathbf{r}^\top \boldsymbol{\Sigma}^{-1} \mathbf{r}) \right], \quad \boldsymbol{\Sigma} = 3 \cdot \mathbf{I}_2; \quad (10)$$

$$L_2(\mathbf{r}) = \max \left(10^{-5}, 0.1 - 0.05 \sqrt{(\mathbf{A}\mathbf{r})^\top \mathbf{A}\mathbf{r}} \right), \quad \mathbf{A} = \begin{pmatrix} 1/5 & 0 \\ 0 & 1/2 \end{pmatrix}; \quad (11)$$

$$L_3(\mathbf{r}, t) = 0.1 \cdot \exp \left[-0.5(\mathbf{r}^\top \boldsymbol{\Sigma}(t)^{-1} \mathbf{r}) \right], \quad \boldsymbol{\Sigma}(t) = 3(t/50 + 1) \cdot \mathbf{I}_2. \quad (12)$$

In the fields above, the maximum value is 0.1 (units are mM) and this peak concentration is at $\mathbf{r} = (0, 0)$. The field L_2 is a static, non-isotropic, linear field, whereas L_3 is a dynamic field: a Gaussian spreading out over time, similar to what one might expect to be produced by a diffusing drop of nutrients. As expected, as long as the stimulus concentrations and their spatial gradients are within the region observed in training, the population distributions show consistency with those produced when simulating using the original full model.

Computational cost savings Computational savings are given empirically here by comparing running times of simulations for both systems. A hundred (100) cells are simulated in each of the ligand fields, for a time $t_{\text{end}} = 500\text{s}$ and a time-step of $\Delta t = 0.05$. Therefore, one million (1000000) iterations of the main while loop in Algorithms 1, 2 are compared in the reported speed-up factor (Table 1). We observe a speed-up factor of ~ 8 , reducing running times from $\sim 460\text{m}$ to $\sim 60\text{m}$. Table 1 reports speed-up factors for each ligand field experiment.

The reported factor values do not include the costs paid for training the GP and producing the training data. It takes $\sim 4\text{m}$ to train GPs for both Ψ_ϕ functions, and $\sim 10\text{m}$ for producing 20000 observations of pCTMC transitions from the original fine system (10000 training points for each Ψ_ϕ function). The relatively low times compared to simulation times, combined with the fact that one only pays this once, upfront, make these costs negligible.

Accuracy evaluation To evaluate how closely results from the abstracted model are compared to the original one, we applied the Kolmogorov-Smirnov (KS) two-sample test [3] to the population distributions of the two models at several time-points in the simulation, as well as to the distributions of running and tumbling duration. We have 100 samples from each population distribution since we simulated 100 cells. However, in the case of ‘Run’ and ‘Tumble’ duration distributions we have ~ 60000 observations from each, because we aggregate observations from the entire trajectory; we choose a random 1000 sample of these to perform the KS test.⁵ In light of these difficulties, a different test which quantifies the distance between the two distributions (e.g. Jensen-Shannon divergence) might be more useful here, but that requires analytic forms of the distributions.

Inspecting Table 1 we find no KS distance higher than 0.2 indicating very similar distributions, as supported by the associated high p-values. The latter do not allow rejecting the null hypothesis with the current sample, which is that the samples originate from the same distribution. An exception is the ‘Tumble’ duration distributions in the L_1 ligand field, where the somewhat higher KS distance of the large sample sizes gives an exaggerated p-value (see footnote 5).

We note how even in the case of the dynamic L_3 field, the resulting population behaviour of the abstracted model is preserved without any additional training necessary. The fact that the original training occurred in a static field does not affect the ability of the abstract model to cope with a dynamic one.

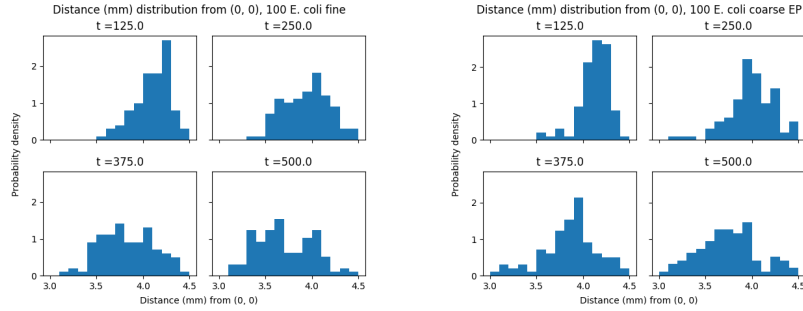


Fig. 4. Empirical distributions for the distance of bacteria populations (100 *E. coli*) at different times t of the simulation. Left: original full system simulations. Right: abstracted system simulations. Gaussian L_1 ligand field.

⁵ We sub-sample because the KS test p-value depends heavily on sample size. Even if two distributions generating samples might be very close, in the limit of an infinite sample size one approaches the true distributions. In such a case, the KS test will reject that the two samples were produced by *the same* distribution, returning lower p-values as sample size increases (for the same KS distance). We do not expect to produce the same distributions here since we are making approximations, so comparing p-values for very large sample sizes is not of interest.

Table 1. KS two-sample test statistics, where the first (top) value reports KS distance and the second (in brackets, bottom) the associated p-value. One sample came from 100 trajectories of fine *E. coli* system simulations, and the other from 100 abstracted system simulations. The first four columns show KS test results of original and abstracted bacterial population distances from peak concentration at various times t (shown in Figure 4). ‘Run’ and ‘Tumble’ columns compare the distributions of run and tumble durations respectively for 1000 samples from each system. The last column reports the observed speed-up factor based on running times and normalising for core utilisation.

Field	$t = 125\text{s}$	$t = 250\text{s}$	$t = 375\text{s}$	$t = 500\text{s}$	Run	Tumble	Speed-up
Gaussian: $L_1(\mathbf{r})$	0.110 (0.556)	0.160 (0.140)	0.170 (0.099)	0.160 (0.140)	0.039 (0.425)	0.101 ($7 \cdot 10^{-5}$)	7.8
Linear: $L_2(\mathbf{r})$	0.010 (0.677)	0.150 (0.193)	0.170 (0.100)	0.130 (0.344)	0.022 (0.967)	0.014 (0.100)	9.4
Dynamic Gaussian: $L_3(\mathbf{r}, t)$	0.140 (0.261)	0.070 (0.961)	0.140 (0.261)	0.080 (0.894)	0.047 (0.214)	0.039 (0.425)	8.9

5 Discussion

In many domains, ranging from cyber-physical systems to biological and medical processes, consideration of spatio-temporal aspects of behaviour is essential. However, this comes at great computational expense. We have presented a methodology that allows layers of a computationally intensive multi-scale model to be replaced by more efficient abstract representations. This is a stochastic map, constructed based on some exploratory simulations of the full model and GP regression. Our results show that we are able to achieve significant speed-up without sacrificing accuracy. This establishes a framework for such statistical abstraction on which we plan to elaborate in future work.

It should be noted that the specifics of the abstraction are not automatically determined by this framework, but are left to the researcher. Having to manually specify the abstraction introduces an element of flexibility, since different abstractions may be tested and so one can see which are suitable and produce accurate approximations, indicating that pertinent elements of the original model are preserved in the coarsening. Additionally, there may be various valid ways to coarsen a model, depending on what the focus of the inquiry is. On the other hand, it shifts some of the burden of abstracting the model to the researcher, who has to find a suitable set of properties which capture the output behaviour of the layer to be abstracted.

Future work avenues include, for example, allowing more properties to be expressed and using them to guide the abstraction will capture more complex behaviours. Additionally, we could infer abstracted model parameters or underlying functions from real data, instead of synthetic ones. Finally, one would like to be able to deal with correlated agents which result in emergent behaviour at the whole population level. This may be readily achieved in this framework if the

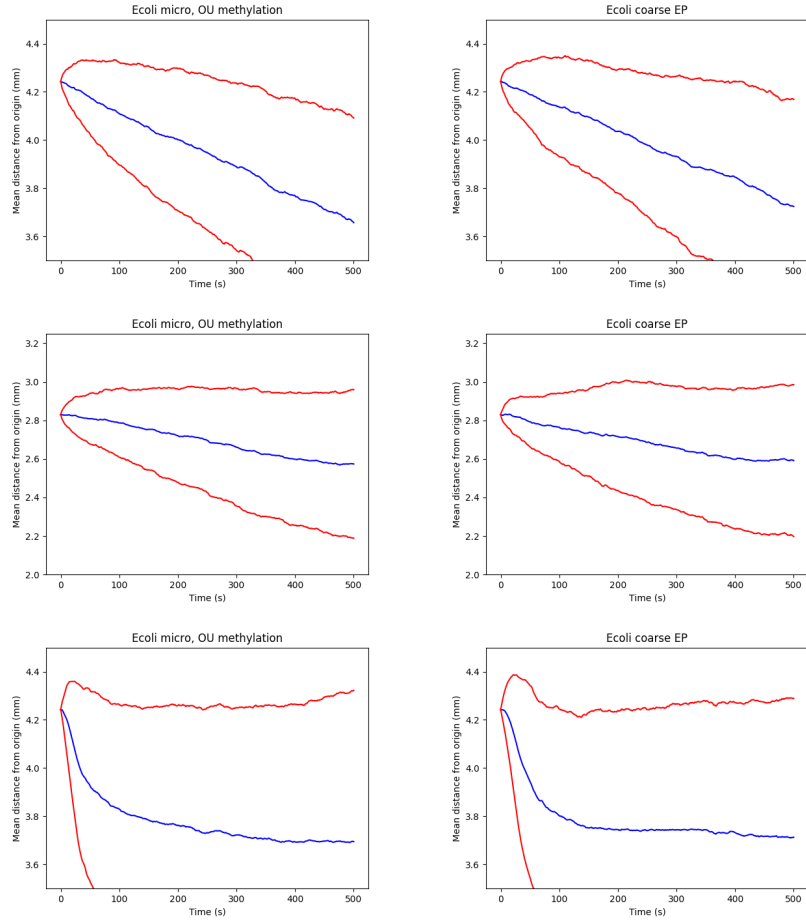


Fig. 5. Average (blue) and standard deviation (red) of distance from peak ligand concentration for a population of 100 *E. coli* over a time of 500s. Left: original full system simulations. Right: abstracted system simulations. Rows (top to bottom): L_1 , L_2 , L_3 ligand fields respectively.

interaction between agents happens by altering their modelled external environment (e.g. by manipulating the nutrient field, or by exuding different chemical trails which can be modelled by an additional external field). However, the path is not so clear if the agents are coupled in some other way, where the internal state of one directly affects that of another.

References

1. L. Bortolussi, D. Milios, and G. Sanguinetti. Efficient Stochastic Simulation of Systems with Multiple Time Scales via Statistical Abstraction. In *Computational Methods in Systems Biology*, pages 40–51. Springer, Cham, 2015.
2. L. Bortolussi, D. Milios, and G. Sanguinetti. Smoothed model checking for uncertain Continuous-Time Markov Chains. *Information and Computation*, 247:235–253, 2016.
3. I. M. Chakravarty, R. G. Laha, and J. D. Roy. *Handbook of methods of applied statistics*. McGraw-Hill, New York, NY, 1967.
4. J. O. Dada and P. Mendes. Multi-scale modelling and simulation in systems biology. *Integrative Biology*, 3(2):86, 2011.
5. N. W. Frankel, W. Pontius, Y. S. Dufour, J. Long, L. Hernandez-Nunez, and T. Emonet. Adaptability of non-genetic diversity in bacterial chemotaxis. *eLife*, 3:e03526, 2014.
6. D. Gilbert, M. Heiner, K. Takahashi, and A. M. Uhrmacher. Multiscale Spatial Computational Systems Biology (Dagstuhl Seminar 14481). 2015.
7. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
8. J. Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *The Journal of Chemical Physics*, 122(18):184102, 2005.
9. C. H. Hansen, R. Endres, and N. Wingreen. Chemotaxis in Escherichia coli: A Molecular Model for Robust Precise Adaptation. *PLOS Comp Bio*, 4(1):e1, 2008.
10. E. L. Haseltine and J. B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of Chemical Physics*, 117(15):6959–6969, 2002.
11. M. Michaelides, D. Milios, J. Hillston, and G. Sanguinetti. Property-Driven State-Space Coarsening for Continuous Time Markov Chains. In *Quantitative Evaluation of Systems*, pages 3–18. Springer, Cham, 2016.
12. J. R. Norris. *Markov Chains*. Cambridge University Press, 1998.
13. C. V. Rao and A. P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *The Journal of Chemical Physics*, 118(11):4999–5010, 2003.
14. C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006.
15. M. W. Sneddon, W. Pontius, and T. Emonet. Stochastic coordination of multiple actuators reduces latency and improves chemotactic response in bacteria. *PNAS*, 109(3):805–810, 2012.
16. E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In Y. Weiss, P. B. Scholkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.
17. V. Sourjik and H. C. Berg. Functional interactions between receptors in bacterial chemotaxis. *Nature*, 428(6981):437–441, 2004.
18. V. Sourjik and N. S. Wingreen. Responding to Chemical Gradients: Bacterial Chemotaxis. *Curr Opin Cell Biol*, 24(2):262–268, 2012.
19. N. Vladimirov, D. Lebiedz, and V. Sourjik. Predicted Auxiliary Navigation Mechanism of Peritrichously Flagellated Chemotactic Bacteria. *PLOS Comp Bio*, 6(3):e1000717, 2010.
20. N. Vladimirov, L. Lvdok, D. Lebiedz, and V. Sourjik. Dependence of Bacterial Chemotaxis on Gradient Shape and Adaptation Rate. *PLOS Comp Biol*, 4(12):e1000242, 2008.